**高级纹理**

**立方体纹理**

**What:**

立方体纹理是环境映射的一种实现方法。立方体纹理包含了6张图像，这些图像对应了一个立方体的6个面。

**Why:**

天空盒和环境映射

**How:**

**立方体纹理的缺点？**

当场景中引入新的物体，光源或者物体发生移动时，我们就需要重新生成立方体纹理。

**如何创建天空盒？**

1. 创建一个材质，设置Shader为Skybox/6 Sided；
2. 使用六张纹理对材质赋值；

**创建用于环境映射的立方体纹理的三种方法。**

1. 直接有一些特殊布局的纹理创建；
2. 手动创建一个Cubemap资源，再把6张图赋给它；
3. 由脚本生成Camera.RenderToCubemap。

**环境映射两个最常见的应用。**

反射和折射

**菲涅尔反射**

**What:**

根据视角方向控制反射程度。

**Why:**

**How:**

**如何计算菲涅尔反射？**

真实世界的菲涅尔等式是非常复杂的，但在实时渲染中，我们使用近似的公式。

Schlick菲涅尔近似等式：

Fschlick(v,n) = F0 + (1 – F0)( 1 - vn)5

**渲染纹理**

**What:**

摄像机的渲染结果可以存到一个中间缓冲中，即渲染目标纹理。

**Why:**

**How:**

**Unity中使用渲染纹理的两种方式。**

1. 在Project目录下创建一个渲染纹理，然后把某个摄像机的渲染目标设置成该渲染纹理；
2. 在屏幕后处理时使用GrabPass命令或OnRenderImage函数来获取当前屏幕图像。

**程序纹理**

**What:**

由计算机生成的图像。

**Why:**

可以通过参数控制纹理的外观。

**How:**

制作程序纹理的第三方软件

Substance Designer